THE FD-68 USER

```
                    AERCO FD-68
 ┌─────────────────────────────────────────┐
 │  ┌──────────┐ ┌────┐                     │
 │                  ┌────────┐        ┌───┐ │
 │    ┌──────────┐  └────────┘        │   │ │
 │    ┌──────────┐ ┌────┐ ┌────────┐ ┌────┐│ │
 │  ┌──────────┐ ┌────┐ │        │ ┌────┐ └─┘│
 │                       └────────┘          │
 │   ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯  ▯                   │
 │                                           │
 │   ▯   ▯ ▯ ▯ ▯ ▯      ▯                    │
 │       ▭▭▭▭▭▭▭▭▭▭▭▭▭▭▭▭▭▭                  │
 └─────────────────────────────────────────┘
```
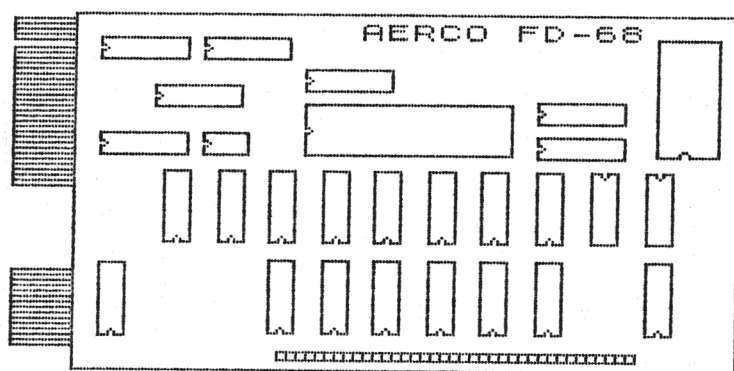
*** Editorial ***

Welcome to the first installment of The FD-68 User newsletter. The purpose of this user's group is to share information and ideas on the T/S 2068 and the AERCO FD-68 disc system.

This will be a different kind of group, in that we don't meet as a group. Response from the ad has shown no concentration of FD-68 users in any one area. This newsletter will be our meeting place. The limited scope of this first issue is due to the fact that it's a solo effort. The quality and quantity of further issues will depend on the amount of input I receive from you. Any ideas, comments, questions or contributions are welcome. I'm confident that this user's group, with its exchange of ideas, will lead to a greater understanding of the T/S 2068, and the FD-68 with its added memory, through software and hardware.

Chunk Enable — OUT 244,(0-255)

The FD-68 system with its 64K, reside in the Dock Bank of the computer. With the .LRO extension not yet implemented, we must limit our use of the extra memory to the upper 32768 bytes. To enable an 8K chunk(s), use OUT 244,(0-255). Be careful of the number you "OUT". The computer will lockup if you enable any of the lower 4 chunks (0-3). If you were to OUT 244,128, you would be enabling chunk 7 of the Dock Bank in preference over the same chunk in the Home bank.

Chunk Boundries 4-7

Chunk 4= 32768-40959
Chunk 5= 40960-49151
Chunk 6= 49152-57343
Chunk 7= 57344-65535

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0
8K	8K	8K	8K	8K	8K	8K	8K

Upper 4 Chunks
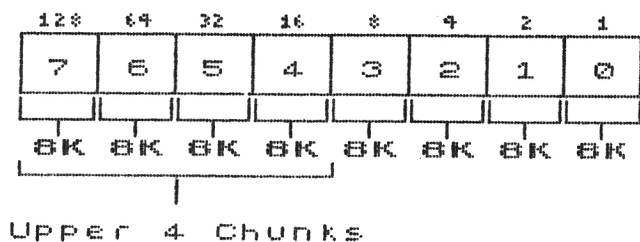
To determine what number to OUT, look at the chart above. To enable a chunk(s) use the small size numbers above the chunk blocks. To enable chunk 5, OUT 244,32. To enable chunk 5 & 6, add 32 (chunk 5) + 64 (chunk 6)= 96. Then OUT 244,96.

Try This in the Immediate mode.

```
OUT 244,0             Enable all of Home Bank
POKE 60000,65         Put 65 at address 60000 in Home Bank
OUT 244,128           Enable Chunk 7 of the Dock Bank
POKE 60000,255        Put 255 at address 60000 in Dock Bank
```

You have now entered, at address 60000 in Home Bank-65.
                                  60000 in Dock Bank-255.

To verify this:

```
OUT 244,0             Enable Home Bank
PRINT PEEK 60000      Returns a value of 65
OUT 244,128           Enable Chunk 7 of th Dock Bank
PRINT PEEK 60000      Returns a value of 255
```

While on the subject of OUT command, I accidently did an OUT 144,1. This caused my 2040 printer to advance one pixel line. After trying different combinations I came up with this.

```
10 FOR a=1 TO 12
20 OUT 144,1
30 PAUSE 2
40 NEXT a
```

Without the PAUSE statement the printer will only advance one line. The PAUSE is needed to slow down the program so the printer can keep up. This will (as close as I can tell) advance the paper the same amount as an LPRINT statement. As far a practical application for this, well that's up to you.

USING SOME OF THAT EXTRA MEMORY

On of the most commonly asked questions is what to do with all that extra memory. What we needed was a way to move data from one bank to the other. I used the block move routine to transfer screen images back and forth. While this was entertaining, it wasn't too practical. I needed something more versatile and useful. Dave McNeely, a fellow 2068 user here in Holland helped me write some machine code to transfer Tasword files as well as the machine code to operate, from one bank to the other and back again. Dave did the original flow chart and steered me in the right direction. Once altered, each time Tasword is loaded in, there will be a 1.5 second pause before executing. The upper 32K of memory is being copied into the Dock bank. This will prevent a crash if you try to read the Dock bank before you have transfered any files there.

Since Tasword's code doesn't fall on any of the 8K chunk boundries, the routine transfers all of the upper 4 chunks (32768 bytes). This is our first try at machine code, but it works OK. It will transfer all 32768 bytes in about 1.5 seconds, one byte at a time. Talk about the speed of machine code!

Lines 60 & 70 were written by Kevin O'Connor of Chicago. There's a lot of programs available that work with Tasword, such as Tasterm, Tasprint, Taswide, etc. Maybe we could call this one TasBank. Nah!-------"Well?"

TASBANK

Here is an attempt to break dowm the machine code used in the TasBank routine. With some changes, this same routine could fit a number of different uses.

```
685B 21FFFF    LD HL,FFFF    Load HL, 65535 (highest mem. addr)
685E 010080    LD BC,8000    Load BC, 32768 (Source/Dest. addr)
6861 F3        DI            Disable system interrupt
6862 0A        LD A,(BC)     Load Accumulator with the value of
                             address pointed to by BC (source)
6863 F5        PUSH AF       Push the contents of accumulator
                             onto the Stack
6864 3EF0      LD A,F0       Load Accumulator with 240
6866 D3F4      OUT (F4),A    Out 244,value held in the Accumul-
                             ator (240)-Enables chunks 4-7 Dock
6868 F1        POP AF        Load Accumulator with Stack value
6869 02        LD (BC),A     Load address pointed to by BC
                             (destination) with value of Acc.
686A 3E01      LD A,01       Load Acc. with 1
686C D3F4      OUT (F4),A    Out 244,1 -Renables Home chunks,
686E 03        INC BC        Increment BC (source/dest.) by 1
686F E5        PUSH HL       Push 65535 on the Stack
6870 ED42      SBC HL,BC     Subtract BC from HL- result in HL
6872 E1        POP HL        Load HL with stack value (65535)
6873 C26268    JP NZ,6862    If 65535-BC<>0 then jump to 6268
6876 FB        EI            Enable system interrupt
6877 C9        RET           Return
```

USING TASBANK

Load Tasword II and exit to basic. Then delete the following lines:
1,25,28,30,35,40,45,50,55,70,150,180,200,500,630,700,710,770, 780,790,900,1020,1030,1110,2000,2010,2020,2030,4000.

The reason for so many line deletes, is to cut out all the tape related commands. I also took out, or trimmed many other lines to save space. Now add the lines listed below. Some replace those deleted, others are additions. If you have more than one drive you can put a drive identifier in the MOVE and CAT commands on lines 1030 and 2030. For example, change CAT "a$", to CAT "b:a$",. In this way you can keep the program itself on one disc and the files on another.

Next, RUN 9000. After it has run, delete lines 9000 and 9010. Now, RUN the program. Exit back to the command menu and save a new copy of Tasword with the "t" option. Ater making a copy to disc, GO TO 15. Now you're all set. Load in your first file. Then transfer it to the Dock bank using the "H" command. Next load in another file. This one will be residing in the Home bank. A screen prompt will inform you as to what bank you're in. Use a "h" to read the Home file, and "d" to read the Dock file. Since all the code is also transfered, Tasword behaves no different than normal. Keep in mind that when you transfer files, you will write over what is currently in the other bank. There will be more on Tasword in the next installment.

```
    1 REM xxxxxxxxxx(37 x's)xxxxx
xxxxxxxxxxxxxxx
   15 POKE VAL "23609",VAL "2": C
LEAR VAL "54783": INK 7: PAPER 0
: CAT "word.bin",: CLS : LET a=U
SR VAL "59081": RANDOMIZE USR VA
L "26715": GO TO VAL "10"
   25 LET in=IN VAL "244"
   30 IF in=VAL "240" THEN LET b$
="Dock"
   35 IF in=VAL "1" THEN LET b$="
Home"
   50 PRINT "(p)rint"'"(s)ave"'"(
j)load"'"(y)read file"'"(g)raphi
cs"'"(t)save tasword"'"(b)exit t
o basic"'" FILE TRANSFER"'"(H)o
me to dock"'"(D)ock to home"''"
READ WHICH FILE"'"(h)ome"'"(d)oc
k"
   52 PRINT ''"Current READ file=
"; FLASH VAL "1"; PAPER VAL "7"
; INK VAL "2"; b$; FLASH VAL "0";
 PAPER VAL "0"; INK VAL "7";" Ba
nk."
   55 PRINT INVERSE 1;AT VAL "20"
,VAL "8";"Press Any Key"
   60 PRINT #0; PAPER 2;AT 0,1;ch
;" Characters used ";((PEEK 6415
0+256*PEEK 64151+22)*64)-ch;" Fr
ee"
   70 PRINT #0; PAPER 2;AT 1,1;a/
64;" Lines Used ";(PEEK 64150+25
6*PEEK 64151+22)-a/64;" Lines Fr
ee"
   90 LET i=VAL "0": LET b=CODE a
$: IF b<VAL "68" THEN LET b=b+VA
L "32"
   115 IF b=VAL "100" THEN LET i=V
AL "7"
   116 IF b=VAL "68" THEN LET i=VA
L "9"
   135 IF b=VAL "104" THEN LET i=V
AL "11"
   136 IF b=VAL "72" THEN LET i=VA
L "13"
   180 IF i>0 THEN GO TO 500
```

```
   200 CLS : PRINT AT VAL "4",VAL
"8";"PRINT OPTIONS"
   320 INPUT "128-143 or ENTER ";a
$
   500 PRINT AT VAL "20",VAL "0";"
ENTER=cont., C=change choice"
   605 IF b=VAL "68" THEN GO TO 71
00
   615 IF b=VAL "72" THEN GO TO VA
L "7300"
   625 IF b=VAL "100" THEN GO TO V
AL "7350"
   630 IF b=VAL "104" THEN GO TO V
AL "7360"
   700 IF in<>VAL "1" THEN CLS : G
O TO VAL "1002"
   705 CLS : LET i=VAL "8": GO SUB
VAL "800": MOVE "boot.bas",15
   710 MOVE "word.bin",54784,10751
: GO SUB VAL "900": GO TO VAL "1
"
   720 LET a$="tasword"
   770 RETURN
1002 IF in<>1 THEN PRINT AT VAL
"10",VAL "0"; INVERSE 1;" MUST B
E IN HOME BANK TO SAVE ": PAUSE
VAL "200": GO TO VAL "20"
1020 IF LEN a$=VAL "0" THEN : CL
S : PRINT AT VAL "12",VAL "5"; I
NVERSE 1;"THERE MUST BE A NAME":
 PAUSE VAL "60": CAT "",: PAUSE
VAL "400": GO TO VAL "1005"
1030 LET i=VAL "12": GO SUB VAL
"800": LET a$=a$+".bin""","+STR$
b+","+STR$ a: MOVE "a$",: CLS
1050 PAUSE 0: GO TO 20
2000 IF in<>1 THEN CLS : PRINT A
T VAL "10",VAL "0"; INVERSE 1;"
MUST BE HOME BANK TO LOAD ": PAU
SE 200: GO TO VAL "20"
2005 CLS : PRINT AT VAL "7",VAL
"1";"Type The Name of The Text F
ile";AT VAL "8",VAL "1";"And Pre
ss ENTER"'","OR"
2010 PRINT AT VAL "12",VAL "1";"
Just Press ENTER"'" for Disc Dir
ectory"
2020 LET j0=VAL "0": LET i=VAL "
16": GO SUB VAL "6000": PRINT AT
 VAL "18",VAL "3"; INVERSE 1;"DI
SC FILE NOW BEING LOADED"
2025 IF a$="" THEN CAT "",: PAUS
E VAL "400": GO TO VAL "2000"
2030 LET b=FN p(VAL "62216"): LE
T a$=a$+".bin""",33280": CAT "a$
",: GO TO VAL "10"
4000 RETURN
7100 POKE VAL "26747",VAL "240":
 RANDOMIZE USR VAL "26745": POKE
 VAL "26725",VAL "1": POKE VAL "
26731",VAL "240": RANDOMIZE USR
VAL "26715": GO TO VAL "20"
7300 POKE VAL "26747",VAL "1": R
ANDOMIZE USR VAL "26745": POKE V
AL "26725",VAL "240": POKE VAL "
26731",VAL "1": RANDOMIZE USR VA
L "26715": GO TO VAL "20"
7350 POKE VAL "26747",VAL "240":
 RANDOMIZE USR VAL "26745": GO T
O VAL "0"
7360 POKE VAL "26747",VAL "1": R
ANDOMIZE USR VAL "26745": GO TO
VAL "0"
9000 RESTORE : FOR a=26715 TO 26
751: READ b: POKE a,b: NEXT a
9010 DATA 33,255,255,1,0,128,243
,10,245,62,240,211,244,241,2,62,
1,211,244,3,229,237,66,225,194,9
8,104,251,201,0,243,62,240,211,2
44,251,201
```

ARTWORX * FD-68

Artworx is already the finest drawing program I've used. Using it with the disc makes it just that much better.

Alter the following lines to read like this:

```
2330 IF M=VAL "1237" THEN GO SUB
 VAL "8350": GO SUB VAL "7800":
GO SUB VAL "8350": PAPER VAL "0"
: INK VAL "0": CLS : CAT "",: IN
PUT "ENTER NAME TO LOAD ";N$: LE
T N$=N$+".SCR"","": CAT "N$",: IN
K VAL "7": PRINT PAPER VAL "8";
INK VAL "9"; AT VAL "21",VAL "10
";"Space=Menu": PAUSE VAL "0": B
EEP VAL ".1",VAL "20": GO TO MEN
U
5565 CAT "",: INPUT "Name this U
DG file ";u$: LET u$=u$+".bin"",
"+STR$ 65368+","+STR$ 168: MOVE
"u$",
5605 GO SUB VAL "5900": GO SUB V
AL "8350": CAT "",: INPUT "Load
which udg file ";u$: LET u$=u$+"
.bin"","+STR$ 65368
5610 CAT "u$",
8675 GO SUB VAL "8350": LET N$=N
~+".SCR"","": MOVE "N$",: GO TO M
ENU
8910 CAT "MC.BIN",
```

To use on a full size printer, merge in your driver software and change the COPY command on line 2315 to whatever your driver software specifies.

Pro/File 2068 * FD-68

Change the end of line 40 of the loader from- LOAD "p/f"CODE 63488,2046, to CAT "pro.bin",.
Change line 50 to CAT "file.bas",.

In the main program alter the following lines to read:
```
107>IF X$="SAVE" THEN CAT "",:
INPUT "ENTER NAME FOR THE FILE "
;F$: LET F$=F$+".BAS"","1": MOVE
"F$": GO TO 1
5510>CAT "",: PRINT AT 16,1; PAP
ER 5; INK 0;"WHAT FILE NAME DO Y
OU WISH    ";AT 17,1;"TO LOAD ";
: INPUT F$: PRINT F$: LET F$=F$+
".BAS": CAT "F$",: GO TO 1
```

As with Tasword, if you have more than one drive you can insert a drive identifier in the first "CAT" command in lines 107 and 5510, e.g. CAT "b:",.
When you save a program by the same name more than once, it will write over the first one. When you save a file from Pro/File you will now be asked for a file name. This is to prevent accidental erase or over-writing of an existing file. I added the additional CAT "", commands on line 107 and 5510 to provide a directory of the current files.

5

How Many ROMs?

Someone I was talking with the other day made the point that we don't know whether or not Timex produced 2068's with more than one version of the ROM. Later, I stumbled across this paragraph in the technical manual.

3.1 Identifier

Location 19 (13H) of the Home Bank ROM is used to identify the revision level of the System Software. The initial version is identified by this location having a value of 255 (FFH). Any subsequent versions will decrement this value by 1, e.g., the first revision would be identified by a value of 254 (FEH). This identifier should be used to conditionally apply patches or execute "work-arounds" identified as necessary with a particular version of the System Software.

Doing a PRINT PEEK 19 on my own system resulted with a 255. If any of you get different results drop me a line.

Many of the books written for the 2068 make mention of the ROM routines. However, they do not list any of them because of the expected ROM revisions by Timex. If we could determine that only one ROM was ever made available, then maybe using ROM routines in our programs would be possible.

As a matter of fact, I have a program on the market that does use a ROM routine. This routine clears the 23rd and 24th lines of the screen. The program has been run on many different 2068s with no problems.

Try This:

```
10 PRINT #1; AT 0,0;"Testing s
creen clear on line 23."; AT 1,0
;"Testing screen clear on line 2
4."
15 PAUSE 0: RANDOMIZE USR 2217
: PAUSE 0: GOTO 10
```

Stray Bytes

Using the ROMSWITCH with the disc system has given me no trouble. It enables me to see my Spectrum programs in RGB color. If you OUT 244,1 in Spectrum mode, you will be greeted with an interesting crash.

I purchased the corrected EXROM from Sinware and found it not to be compatible with my disc system. AERCO says that some people have trouble and others don't. I put the EXROM in a 2068 without the disc system and it worked as advertised. It just didn't like my disc.

I spoke with AERCO on 12/27 and they said that the CP/M would be available soon. As to when the ".CHR" and ".DAT" extensions would be implemented, they said it would follow soon after the CP/M was done. Good things from AERCO come to those who wait.

AERCO is also working on a hardware change that would allow the use of cartridges without disconnecting the interface. In the mean time, AERCO is willing to transfer the contents of your cartridges to disc for a small fee.

"OUT" REVISITED

Just one last bit before signing off. When you use the OUT 244,(0-255) command, you are writing to a port. Sort of like POKEing an address. PRINT IN 244, will return the value last written to that port, in the same way as PRINT PEEK returns the value last written to that address. Since we use port 244 to enable the different chunks of memory in the Dock bank, we can use IN 244 in a conditional statement to determine which chunks have been enabled and proceed accordingly.

Well that does it for this time around. Even if you didn't learn a thing from it, maybe it sparked an idea from which great things will come. The information contained herein is accurate to best of my knowlege. If you spot some errors please let me know. Mistakes make great teachers (if you learn from them).

This newsletter was written using Tasword II, Artworx, and an Epson LX-80 printer. After writing this newsletter, as short as it is, I no longer take for granted the other publications out there. Not being a writer, this newletter took longer to prepare than I first realized. I appreciate your patience in waiting for this first installment.

I'm tentatively planning on 4 issues per year. If the volume of articles and such are sufficient, maybe more often. Right now it's a wait and see proposition.

Please send any questions, ideas, comments, answers, contributions etc. to:

The FD-68 User
David A. Hill
1159 South Shore Drive #12
Holland, MI   49423

Have a question or idea that can't wait? I'd be glad to talk with any of you on the phone, verbally or with a modem. Call after 5.00 P.M. (616) 335-8726.